

## **REMARKS**

Claims 1-53 are pending in the case. The Office Action rejected each of claims 1-53 as follows:

- claims 1-12, 15-24, 27-33 and 38-53 as being anticipated under 35 U.S.C. §102(b) by U.S. Letters Patent 6,493,868 (“DaSilva et al.”);
- claims 13, 14, 25 and 26 as obvious under 35 U.S.C. §103(a) by DaSilva in combination with U.S. Publication No. 2002/0087916 (“Meth”); and
- claims 34-37 as obvious under 35 U.S.C. §103(a) by DaSilva in combination with U.S. Publication No. 2002/0016953 (“Sollich”).

The Office also objected to claim 12 for an informality. Applicants traverse each of the rejections.

### **I. INFORMALITIES**

The Office did not state whether the drawings have been accepted. Since there was no objection, Applicants presume they have been and request notification if in error.

The Office objected to claim 12 for a “surplus word”. Applicants have amended claim 12 to cure this objection. This amendment therefore was not made for purposes of patentability. Furthermore, it does not narrow the scope of the claim.

### **II. RESPONSE TO SUBSTANTIVE MATTERS**

#### **A. CLAIMS 1-12, 15-24, 27-33 AND 38-53 ARE NOVEL**

The Office rejected claims 1-12, 15-24, 27-33 and 38-53 as being anticipated under 35 U.S.C. §102(b) by U.S. Letters Patent 6,493,868 (“DaSilva et al.”). Each of the independent claims 1, 12, 18, 22, 24, 28, 38, 44, and 50 recites a “waypoint” and performing some act defined relative to such a waypoint. Each of the dependent claims 2-11, 15-17, 19-21, 23, 27, 29-33, 39-43, 45-49, and 51-53 incorporates this limitation as a matter of law by virtue of their dependence. 35 U.S.C. §112, ¶4.

To anticipate, DaSilva et al. must disclose every limitation of the rejected claim in the same relationship to one another as set forth in the claim. M.P.E.P. § 2131; *In re Bond*, 15

U.S.P.Q.2d (BNA) 1566, 1567 (Fed. Cir. 1990). DaSilva et al. fails to do this if it is construed properly—that is, by one of ordinary skill in the art.

Consider, for example, the rejection of claim 1. Claim 1 recites “compiling at least a portion of a source code program defined by a waypoint during the editing of the source code program.” The Office cites col. 2, lines 50-55, for teaching this entire claim. This passage reads:

The IDE can create a full build, *an incremental build*, and can compile individual files. It can also scan files to build a dependency tree for the entire project.

#### Debugging Applications

The IDE supports the following program debugging features: *breakpoints*, with a number of stepping options:

(emphasis added) The Office does not actually map this language onto the claim language, but presumably equates “incremental build” with “compiling at least a portion of a source code program” and “breakpoints” with “waypoints”.

This is not the way one skilled in the art would construe DaSilva et al.. For example, even the most casual of programmers are quite familiar with the term “breakpoint” and would not call it a “waypoint”, and would understand that it refers to a completely different part of the process than the term “compilation”. The term “breakpoint” has a precise meaning in software development environments and a “waypoint” as used by Applicants does not meet that meaning. Furthermore, the concepts of “incremental build” and “executing to a breakpoint” are completely orthogonal concepts. The former refers to the process of constructing the executable program, while the latter refers to the process of running that program).

It is like the difference between building an automobile vs. driving that automobile. No technique for speeding up a factory assembly line producing a car would be confused with a technique for driving the car. That distinction is quite apparent even to laymen. Unfortunately, the area of software development tends to be less approachable, and that distinction less apparent.

More particularly, the term “incremental build” means, in standard developer tools industry parlance, that making a small change to the source code does not necessitate a full recompile of every source file that comprises the software under development. For example, consider a program consisting of 100 source files, and in which one makes a change to a single

source file. An "incremental build" feature means that only the one source file that was changed needs to be recompiled, and not all 100 files (which would be called a "full" build).

After a program has been built (by any means: "full build", "incremental build"), it can be run. Even the most basic of software development environments provides a way to set "breakpoints", which are locations in the build at which execution should suspend. Note, however, that the breakpoints have nothing to do with compilation, but rather with execution.

Compilation, by its very name, refers to the process of "compiling" the source code to produce program code, not the process of running that code once it has been produced. So anything that happens after the program has been built (such as for example running the program, whether to a breakpoint or not) is by definition independent of the process compilation.

Returning to the automobile analogy, "building the program" (which involves compilation, among other things) is analogous to building an automobile on a factory assembly line, while "executing to a breakpoint" is analogous to driving that automobile on the road and coming to a stop sign. The two have no connection, except that obviously the car must be built before it can be driven; similarly, the program must be built before it can be executed.

Thus, DaSilva et al.'s incremental build is not in any way related to its use of breakpoints, since the former is a part of compilation and the latter is a part of execution. Even if one equates the incremental build to "compiling at least a portion of a source code program", that source is cannot be defined by a breakpoint. DaSilva et al. therefore does not teach a "waypoint". Still further, DaSilva et al.'s incremental build does not teach compilation "during the editing of the source code program". DaSilva et al. consequently fails to anticipate claim 1, and any claim dependent therefrom, for several reasons.

Each of the other independent claims 12, 18, 22, 24, 28, and 38 recites a "waypoint". As is established immediately above, the "breakpoints" of DaSilva et al. cannot be the claimed "waypoint" because "breakpoints", as that term is used in the art, are employed in *execution*. The claimed "waypoint", however, is used during *compilation*, not execution. Accordingly, DaSilva et al. fails to anticipate any one of claims 12, 18, 22, 24, 28, and 38 or any claim depending therefrom.

Independent claims 44 and 53 recite compilation during editing. As established above, DaSilva et al. does not teach this limitation. The "incremental build" to which the Office points is something completely different, since it necessarily implies a build *after* editing is complete.

With respect to claim 44, the Office cites col. 8, lines 61-67, which discusses stepping through a source program *during execution, not compilation*. With respect to claim 53, the Office cites col. 9, lines 7-9, which also stepping through a source program *during execution, not compilation*. In fact, this portion actually refers to “running the program”, as opposed to “building” or “compiling” the program. DaSilva et al. therefore fails to anticipate either claim 44 or claim 53, or any claim dependent therefrom.

Thus, DaSilva et al. fails to anticipate any of claims 1-12, 15-24, 27-33 and 38-53. M.P.E.P. § 2131; *In re Bond*, 15 U.S.P.Q.2d (BNA) 1566, 1567 (Fed. Cir. 1990). Applicants therefore request that the rejections be withdrawn.

**B. CLAIMS 13, 14, 25 AND 26 ARE UNOBVIOUS OVER DASILVA ET AL. & METH**

The Office rejected claims 13, 14, 25 and 26 as obvious under 35 U.S.C. § 103(a) by U.S. Letters Patent 6,493,868 (“DaSilva et al.”) in combination with U.S. Publication No. 2002/0087916 (“Meth”). To establish a *prima facie* case of obviousness, the prior art reference (or references when combined) must teach or suggest all the claim limitations. M.P.E.P. § 706.02(j); *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (CCPA 1974). Each of these rejections relies upon DaSilva et al. to teach all the limitations of the claims from which these claims depend. As established above, DaSilva et al. does not do this when properly construed. DaSilva et al. and Meth therefore do not teach all the limitations of these claims. Accordingly, claims 13, 14, 25, and 26 are not obvious over DaSilva et al. and Meth.

**C. CLAIMS 34-37 ARE UNOBVIOUS OVER DASILVA ET AL. & SOLLICH**

The Office rejected claims 34-37 as obvious under 35 U.S.C. § 103(a) by U.S. Letters Patent 6,493,868 (“DaSilva et al.”) in combination with U.S. Publication No. 2002/0016953 (“Sollich”). To establish a *prima facie* case of obviousness, the prior art reference (or references when combined) must teach or suggest all the claim limitations. M.P.E.P. § 706.02(j); *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (CCPA 1974). These rejections rely upon DaSilva et al. to disclose a “waypoint” and “invoking a compile of at least a portion of a source code program defined by a waypoint” as recited in claim 34. As established above, DaSilva et al. fails to teach

either one of these things. Claims 34-37 therefore are not obvious over DaSilva et al. and Sollich.

### **III. CONCLUDING REMARKS**

Applicants therefore respectfully submit that the claims are in condition for allowance, and requests that they be allowed to issue. The Examiner is invited to contact the undersigned attorney at (713) 934-4053 with any questions, comments or suggestions relating to the referenced patent application.

Respectfully submitted,

Date: January 21, 2008

WILLIAMS, MORGAN & AMERSON  
10333 Richmond Dr., Suite 1100  
Houston, Texas 77042  
(713) 934-4053 ph

/Jeffrey A. Pyle/

Jeffrey A. Pyle  
Reg. No. 34,904  
Attorney for Applicants